MRN107

Reconstructing Temperature History From temperature-dependent Degradation Data

Bill Menke, April 2, 2012

Suppose chemical species k degrades (reacts) at a rate $r[k,T(t)]$, which is a function of temperature $T(t)$, where t is time. The rate of change of concentration $C^{(k)}$ of the chemical species is assumed to obey:

$$\frac{dC^{(k)}}{dt} = -r[k,T(t)]\,C \tag{1}$$

The concentration at time t_1 constitute the data $d^{(k)}$:

$$d^{(k)} = -\ln\frac{C^{(k)}}{C_0^{(k)}} = \int_0^{t_1} r[k,T(t)]\ \mathrm{d}t \tag{2}$$

We now ask what can be learned about the temperature $T(t)$ given measurements of $d^{(k)}$ at fixed t_1 but for several (say N) k's. Evidentially, we cannot determine $T(t)$, since the integral in (2) is not sensitive to the time sequence of T. Thus, for instance, if $T(t)$ were piecewise constant, the integral is not sensitive to the order of the pieces. However, (2) is sensitive to the amount of time spent near each temperature, as can be seen by transforming the integral from t to T. Suppose, for the moment, that $T(t)$ is a monotonically increasing function of t. Then

$$d^{(k)} = \int_0^{t_0} r[k,T(t)]\ \mathrm{d}t = \int_{T_0}^{T_1} r[k,T]\,\frac{\mathrm{d}t}{\mathrm{d}T}\mathrm{d}T = \int_{T_0}^{T_1} r[k,T]\ \mathrm{d}t(T) \tag{3}$$

Here $\mathrm{d}t(T)$ represents the increment of time spent near temperature T. If $T(t)$ is not monotonically increasing, then we must break the t-axis into several segments, each in which $T(t)$ monotonically increases or decreases. Equation (3) then becomes

$$d^{(k)} = \sum_i \int_{T_{min}}^{T_{max}} r[k,T]\left|\frac{\mathrm{d}t}{\mathrm{d}T}\right|\mathrm{d}T = \int_{T_{min}}^{T_{max}} r[k,T]\ \mathrm{d}t^{total}(T) \ \text{ where } \ \mathrm{d}t^{total}(T) = \sum_i \mathrm{d}t(T) \tag{4}$$

where the sum is over the segments. We now understand $\mathrm{d}t^{total}(T)$ to mean the total amount of time spent at temperature T. The absolute value sign in (4) is added to indicate that all time increments $\mathrm{d}t(T)$ are considered positive, regardless of whether $T(t)$ is increasing or decreasing in the segment. We now consider the discrete approximation of (4), where we divide the T-axis into M intervals between T_{min} and T_{max}.

$$d^{(k)} = \int_{T_{min}}^{T_{max}} r[k,T]\ \mathrm{d}t^{total}(T) = \sum_{j=1}^{M} r[k,T_j]\ \Delta t^{(j)}$$

(5)

Here $\Delta t^{(j)}$ is the total amount of time spent at temperature T_j. Equation (5) is a standard linear inverse problem of the form

$$\mathbf{d} = \mathbf{Gm} \quad \text{with} \quad d_k = d^{(k)} \ \text{ and } \ G_{kj} = r[k,T_j] \ \text{ and } \ m_j = \Delta t^{(j)}$$

(6)

and can be solved using generalized least squares. The behavior of the solution will depend upon the structure of the matrix **G**, for example, on whether its columns are linearly independent.

Case 1: The rate function r is a linear function of temperature (with no additive constant).

$$G_{kj} = r[k,T_j] = b_k T_j$$

(7)

This problem is completely non-unique, since every column is linearly dependent on every other. Thus for columns p and q:

$$\frac{G_{kq}}{T_q} = \frac{G_{kp}}{T_p} = b_k$$

(8)

Case 2: The rate function r is a linear function of temperature (with non-zero additive constant).

$$G_{kj} = r[k,T_j] = a_k + b_k T_j$$

(9)

This problem is unique, as long as the a_k's are distinct. On the other hand, if four columns (p, q, r, s) all have the same a's, then the problem is substantially non-unique, since linearly -dependent combinations can be formed from pairs of columns.

$$\frac{\left(\frac{G_{kp}}{a} - \frac{G_{kq}}{a}\right)}{(T_p - T_q)} = \frac{\left(1 + \frac{b_k}{a}T_p\right) - \left(1 + \frac{b_k}{a}T_q\right)}{(T_p - T_q)} =$$

$$\frac{\left(\frac{G_{kr}}{a} - \frac{G_{ks}}{a}\right)}{(T_r - T_s)} = \frac{\left(1 + \frac{b_k}{a}T_r\right) - \left(1 + \frac{b_k}{a}T_s\right)}{(T_r - T_s)} = \frac{b_k}{a}$$

(10)

Since (9) can be considered a Taylor series approximation of a general rate function, we conclude that, except in special cases such as the one considered in (10), the general problem is unique.

Case 3: The rate function r is a exponential function of temperature:

$$G_{kj} = r[k, T_j] = a_k \exp{(b_k T_j)} \tag{11}$$

Note that for temperatures near T_0, the linear approximation is

$$G_{kj} = a_k \exp{(b_k T_j)} \approx a_k \exp{(b_k T_0)} + a_k b_k \exp{(b_k T_0)}(T_j - T_0)$$

$$= a_k(1 - b_k) \exp{(b_k T_0)} + a_k b_k \exp{(b_k T_0)} T_j = a'_k + b'_k T_j$$

$$\text{with } a'_k = a_k(1 - b_k) \exp{(b_k T_0)} \text{ and } b'_k = a_k b_k \exp{(b_k T_0)} \tag{11}$$

The a'_k are distinct, so the solution is unique. We examine a numerical example below, with $N=M=11$ and a's and b's as shown in Figure 1. The data are shown in Figure 2. They correspond to the temperature history shown in Figure 3. The reconstruction (green circles in Figure 3) is excellent (though some tuning of the least-squared damping coefficient was needed to achieve this quality of solution).

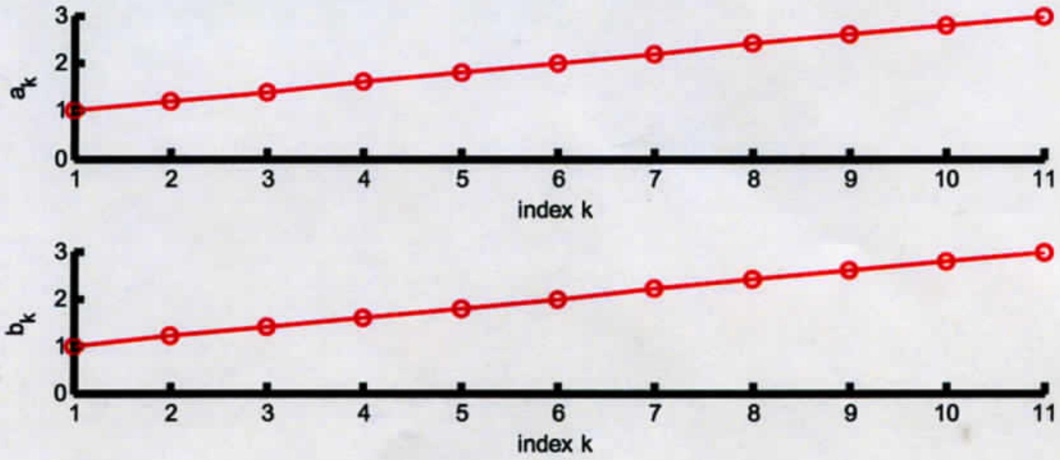

Figure 1. a_k and b_k used in the numerical example.

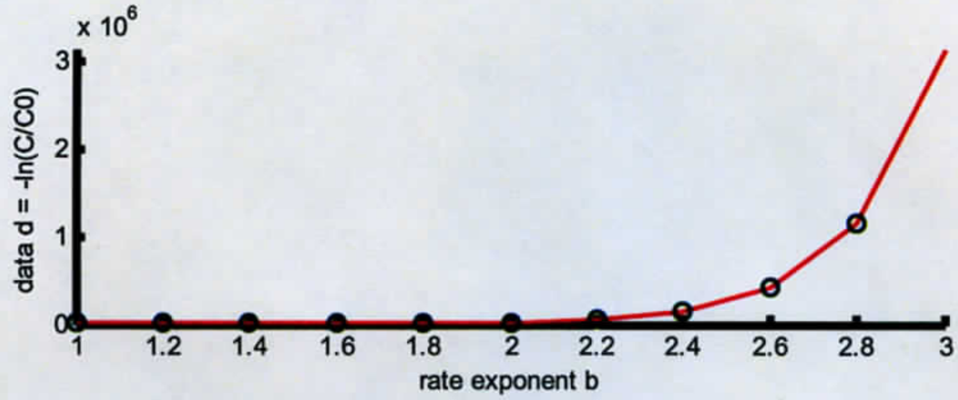

Figure 2. True data (red) based on the a_k and b_k shown in Figure 1 and the temperature history shown in Figure 3. The predicted data (green), based on the estimated solution in Figure 3, closely matches the true data.

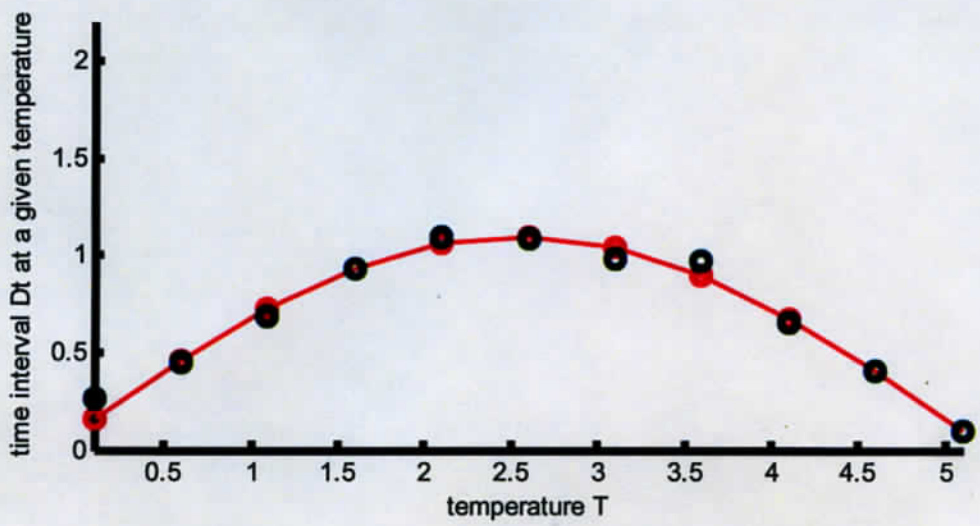

Figure 3. True (red) and estimated (reconstructed) (green) temperature history.

Matlab Script

```
M=11;
DT = 0.5;
T = 0.1+DT*[0:M-1]';
Tmin = min(T);
Tmax = max(T);

dt_true = 0.1+sin( pi*T/Tmax );
dt_max = max(dt_true);
```

```
figure(1);
clf;
set(gca,'LineWidth',3);
hold on;
axis( [Tmin, Tmax, 0, 2*dt_max] );
plot( T, dt_true, 'ro', 'LineWidth', 3);
plot( T, dt_true, 'r-', 'LineWidth', 2);
xlabel('temperature T');
ylabel('time interval Dt at a given temperature');

C0 = 1;

N = 11;
Da = 0.2;
a = Da*[0:N-1]'+1;
Db = 0.2;
b = Db*[0:N-1]'+1;

G=zeros(N,M);
for i=[1:N]
for j=[1:M]
    G(i,j) = a(i)*exp(b(i)*T(j));
end
end
d_true = G*dt_true;
d_max = max(d_true);

figure(2);
clf;
set(gca,'LineWidth',3);
hold on;
axis( [b(1), b(N), 0, d_max] );
plot( b, d_true, 'r-', 'LineWidth', 2);
xlabel('rate exponent b');
ylabel('data d = -ln(C/C0)');

s2=0.1;
d_obs = d_true + random('Normal',0,s2,N,1);
plot( b, d_obs, 'ro', 'LineWidth', 2);
```

```
e2=1.0e-1;
dt_est = (G'*G + e2*eye(M))\(G'*d_obs);
figure(1);
plot( T, dt_est, 'go', 'LineWidth', 3);

d_pre = G*dt_est;
figure(2);
plot( b, d_obs, 'go', 'LineWidth', 2);


figure(3);
clf;
subplot(2,1,1);
set(gca,'LineWidth',3);
hold on;
axis( [1, N, 0, a(N)] );
plot( [1:N]', a, 'r-', 'LineWidth', 2);
plot( [1:N]', a, 'ro', 'LineWidth', 2);
xlabel('index k');
ylabel('a_k');

subplot(2,1,2);
set(gca,'LineWidth',3);
hold on;
axis( [1, N, 0, b(N)] );
plot( [1:N]', b, 'r-', 'LineWidth', 2);
plot( [1:N]', b, 'ro', 'LineWidth', 2);
xlabel('index k');
ylabel('b_k');
```